Attacks on Implementations of Secure Systems

May 2, 2019

Chapter 1

Introduction

Once upon a time... This document shows how you can get ePub-like formatting in LaTeX with the `memoir` document class. You can't yet export directly to ePub from writeLaTeX, but you can download the source and run it through a format conversion tool, such as `htlatex` to get HTML, and then go from HTML to ePub with a tool like Sigil or Calibre. See `http://tex.stackexchange.com/questions/16569` for more advice. And they lived happily ever after.

Contents

Chapter 2

Writing LaTeX

2.1 Basic Formatting

Comments. If you want to just add a comment to a file without it being printed, add a `%` (percentage) sign in front of it. In the template files, you will find a number of such comments as well as deactivated commands.

Bold formatting. You can make your text bold by surrounding it with the command `\textbf{}`.

Italics formatting. You can make your text italic by surrounding it with the command `\textit{}`.

Small caps. You can change your text into small capitals by surrounding it with the command `\textsc{}`.

Text em dashes. Em dashes are used to connect two related sentences. There is no space before or after the em dash. Within the template, use the command `\textemdash{}` instead of using the dash you copied over from your text file. This will also take care of issues relating to line breaks.

Paragraphs. Paragraphs are handled automatically by leaving an empty line between each paragraph. Adding more than one empty line will not change anything—remember it is not a "what you see is what you get" editor.

Empty line. If you want to force an empty line (recommended only in special cases), you can use `~\\` (tilde followed by two backslashes).

New page. Pages are handled automatically by LaTeX. It tries to be smart in terms of positioning paragraphs and pictures. Sometimes it is necessary to add a page break, though (ideally, at the very end when polishing the final text). For that, simply add a `\newpage`.

Quotation marks. In the normal computer character set, there are more than one type of quotation marks. It

is required to change all quotation marks into ``...'' (two back ticks at the beginning and two single ticks at the end) and refrain from using "..." (or "...") altogether. This is because Word's "..." uses special characters, and "..." do not mark the beginning and end of the quotation.

Horizontal line. For a horizontal line, simply write `\hline`.

Underlined text. It is generally not recommended to use underlined text.

URLs. For URLs you need a special monospaced font. Also, for URLs in e-books, you want to make them clickable. Both can be accomplished by putting the URL in the `\url{}` environment, for example `\url{https://www.lode.de}`.

Special characters. If you need special characters or mathematical formulas, there is a whole body of work on that subject. It is not in the scope of this book to provide you a comprehensive list.

2.2 Lists

Itemized list. To create a bullet point list (like the list in this section), use the following construct:

```
\begin{itemize}
    \item Your first item.
    \item Your second item.
    \item Your third item.
%   \item Your commented item.
\end{itemize}
```

The result will look like this:

- Your first item.
- Your second item.
- Your third item.

Numbered list. To create a numbered list, replace itemize with enumerate:

```
\begin{enumerate}
    \item Your first item.
    \item Your second item.
    \item Your third item.
\end{enumerate}
```

The result will look like this:

1. Your first item.

2. Your second item.

3. Your third item.

2.3 Verbatim text

Sometimes, you do want to simply use text in a verbatim way (including special characters and LaTeX commands). For this, simply use the `\lstlisting` environment: `\begin{lstlisting}...\end{lstlisting}`. For example, I put the itemize and enumerate listings above into a `\lstlisting` block. If I did not, LaTeX would have displayed the list as a list, instead of displaying the code.

2.4 Chapters and Sections

LaTeX uses a hierarchy of chapters, sections, and subsections. There are also sub-subsections, but for the sake of the reader, it is best to not go that deep. If you come across a situation where it looks like you need it anyway, I recommend thinking over the structure of your book rather than using sub-subsections.

In terms of their use in the code, they are all similar:

- `\chapter{Title of the Chapter}\label{c1_chaptername:cha}`

- `\section{Title of the Section}\label{c1_sectionname:sec}`
- `\subsection{Title of the Subsection}\label{c1_subsectionname:sec}`
- `\paragraph{Title of the Paragraph}\label{c1_paragraph:sec}`

When using these commands, obviously replace the title, but also the label. For the label, I recommend to have it start with c, followed with the current chapter number, an underscore, and the chapter, section, or subsection in one word and lowercase, followed by either ":cha" or ":sec" to specify what kind of label it is. These labels can then be used for references like we used previously for the images. For example, if you have defined a section `\section{Chapters and Sections}\label{c1_chaptersandsections:sec}`, you could write "We will discuss chapters and sections in section `\ref{c1_chaptersandsections:sec}` which results in the document in "We will discuss chapters and sections in section 2.4".

2.5 Tables

In LaTeX, tables are like images and put into the figure environment. As such, they have a caption, label, and

a positioning like we discussed above with the images. Drawing a table requires a bit of coding:

```
\begin{table}[!ht]
    \centering
    \begin{tabular}{p{2.5cm}|p{3.5cm}|p{3.5cm}}
    \hline
    & \textbf{Word} & \textbf{\LaTeX{}} \\
    \hline

    Editor & ``what you see is what you get'' & source file is compiled \\
    \hline

    Compatibility & dependent on editor & independent of editor \\
    \hline

    Graphics & simple inbuilt editor & powerful but complex editor \\
    \hline

    Typography & optimized for speed & optimized for quality \\
    \hline
```

```
19
20      Style & inbuilt style & separate↩
            style document \\
21      \hline
22
23      Multi-platform & only via export↩
            & possible with scripting \\
24      \hline
25
26      Refresh & some elements need, ↩
           manual refresh & everything ↩
           is refreshed with each ↩
           compile \\
27      \hline
28
29      Formulas & basic support needs ↩
           external tools & complete ↩
           support \\
30      \hline
31
32      \end{tabular}
33      \caption{Comparison of Word and ↩
           \LaTeX{}} \label{c1_↩
           comparisonwordlatex:tab}
34  \end{table}
```

This table from the beginning of the book has the familiar figure, label, caption, and centering commands. The actual table is configured with the `\tabular{}` envi-

ronment. Following the tabular command, you configure the columns in curly braces. Each column is separated with a vertical line and the `p{...}` entry specifies the width of the column. With `{p{2.5cm}|p{3.5cm}|p{3.5cm}}`, you would have three columns with 2.5cm width for the first column and 3.5cm width for the two others. Alternatively, you can use `c` instead of `p` and leave out the curly braces with the width. Then, LaTeX simply calculates the required widths automatically. Then, for each line of the table, simply write: `content of the first cell & content of the second cell & content of the third cell\\\hline`.

2.6 Footnotes

Finally, for footnotes, there is the command `\footnote{}`. You can place it anywhere you like, LaTeX will then automatically add the number of the footnote at that place, and put the footnote text into the footer area. It looks like this.[1] The challenge here relates to grammar: footnotes start with capital letters, parentheses with lower case, and the footnote comes after the period, the parentheses have to start before the period.

[1] This is a footnote.

	Word	**LaTeX**
Editor	"what you see is what you get"	source file is compiled
Compatibility	dependent on editor	independent of editor
Graphics	simple inbuilt editor	powerful but complex editor
Typography	optimized for speed	optimized for quality
Style	inbuilt style	separate style document
Multi-platform	only via export	possible with scripting
Refresh	some elements need, manual refresh	everything is refreshed with each compile
Formulas	basic support needs external tools	complete support

Table 2.1: Comparison of Word and LaTeX

Chapter 3

Inserting Images

3.1 Images

As in Word, in LaTeX, images are separate from the text. Images are usually packaged together with a caption and a label to reference it from the text. These three entities are packaged together into a figure. The figure itself configures the size of the image as well as where it should be put. Let us look at a code sample:

```
1  \begin{figure}[H]
2      \centering
3      \includegraphics{images/↩
           ebookLatex_Cover.jpg}
4      \caption{The cover of this book↩
           .} \label{c1_cover:fig}
5  \end{figure}
```

Let us go through this line by line. At the core is the image, included with `\includegraphics{path to file}`. It inserts the image specified by the "path to file." With the `\adjustbox{}` command, we can adjust the image size according to the page width (`\columnwidth`) and page height (`\textheight`).

Below there is the caption and the label. LaTeX automatically numbers each figure, so in the text, we can later refer to it with `\ref{c1_cover:fig}` which prints out the number of the figure. Finally, all these commands are centered with the `\centering` command and surrounded with the figure environment. The `[ht]!` instructs LaTeX to try to place the image exactly where it is in the LaTeX code.

In Figure 3.1, you can see the result of the command. Instead of graphics, you can also include other TEX files that contain graphics (or commands to draw graphics, see chapter 3.2).

3.2 TikZ Graphics

For graphics, you can use the inbuilt TikZ graphics generator. Due to its flexibility, I even recommend images you already have for a number of reasons:

Figure 3.1: The cover of this book.

- TikZ graphics can very easily changed (especially for for example translations or making corrections).
- TikZ graphics are small and flexible. They can be

easily scaled to any size and are directly integrated into your project (no time-consuming editing in an external graphics program necessary).

- TikZ graphics look better. As vector graphics are sent directly to the printer, we need not to worry about readability.

If you want to create a TikZ graphic, simply create a new TEX file in the *tex-images* folder and include it with `\input` (replacing `\includegraphics{}`) where you want to.

Then, do a "recompile from scratch" by clicking on the top right corner of the preview window (showing Warning or Error) to regenerate the TikZ file. If "up-to-date and saved" is shown, delete the *tikz-cache* directory and recreate it.

For the format of the file itself, it is a series of commands surrounded by the `\begin{tikzpicture}...\end{tikzpicture}` environment. Discussing all the commands is beyond the scope of this book, so I recommend three options:

- Check out the PGF manual at `https://www.ctan.org/pkg/pgf`. It is more than 1100 pages full with documentation of each command and corresponding examples.

- Check out the few example TikZ pictures from my two books [1] and [2] in the *tex-images* directory.

Chapter 4

Replace with Third Chapter Name

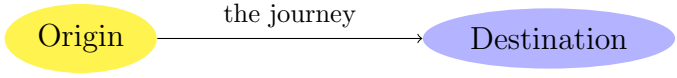

Figure 4.1: TikZ drawings will be output as SVG, which should be rendered by most modern browsers.

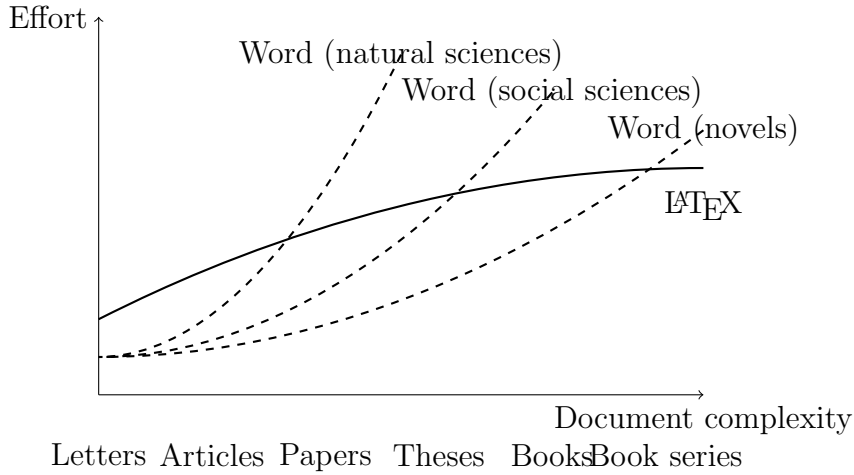

Figure 4.2: Comparing complexity of *Word* and LaTeX depending on the application.

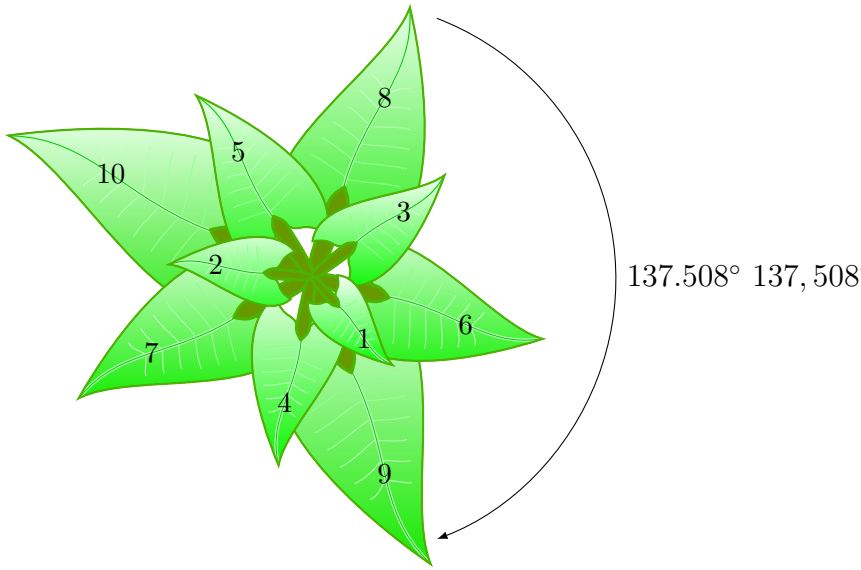

Figure 4.3: Example of a drawing made in TikZ.

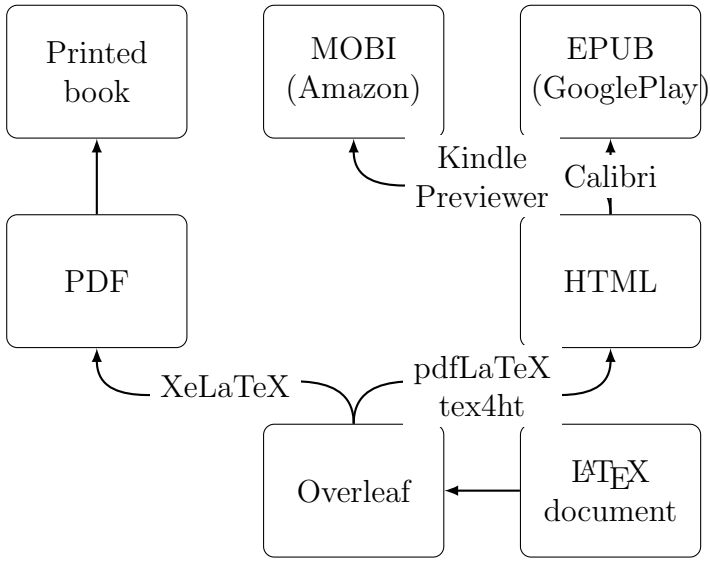

Figure 4.4: Example 2 of a drawing made in TikZ.

Bibliography

[1] Clemens Lode. *Philosophy for Heroes: Knowledge.* Clemens Lode Verlag e.K., 2016.

[2] Clemens Lode. *Philosophy for Heroes: Continuum.* Clemens Lode Verlag e.K., 2017.